Cognitive Radio Engine Learning Adaptation

Olaleye, Martins; Dahal, Keshav; Pervez, Zeeshan

DOI:
10.1109/SKIMA.2016.7916241

Published: 04/05/2017

Document Version
Peer reviewed version

Cognitive Radio Engine Learning Adaptation

Martins Olaleye
School of Engineering and Computing
University of the West of Scotland,
United Kingdom
Email: martins.olaleye@uws.ac.uk

Keshav Dahal
School of Engineering and Computing
University of the West of Scotland,
United Kingdom
Email: keshav.dahal@uws.ac.uk

Zeeshan Pervez
School of Engineering and Computing
University of the West of Scotland,
United Kingdom
Email: zeeshan.pervez@uws.ac.uk

***Abstract*—Cognitive radio (CR) system being an intelligence-based communication device has been considered as the next generation emerging technologies to Wireless Communication Systems (WCS). This CR's embedded-intelligent agent is called Cognitive Engine (CE), and is responsible for the dynamic adaptation between the WCS's environment and the radio operational parameters. As a result of CR's intelligence capability, the WCS's quality of service (QoS) and its connectivity operations get enhanced. In order to evaluate the CR engine performance in respect to its learning, timing, and its computational performances. This paper proposes an alternative state-of-the-art enhanced CR learning engine based on Random Neural Network (RNN). Unlike Artificial Neural Network (ANN) systems, RNN establishes strong data generalization, converges faster and produces relatively smaller levels of prediction errors. Subjected to similar environmental conditions, the simulation cumulative results show that the performance of the proposed RNN system is satisfactory and produces 36.895% performance improvement above the ANN learning engine.**

***Index Terms*—Cognitive Radio, Cognitive Engine, Random Neural Network, Learning, Network simulation**

I. Introduction

Wireless Communication System (WCS) environment is presently being considered to be heavily congested and noisy. This is the magnitude of huge demands observe in today's wireless communication services and applications, such has high mobility, users explosion, outburst in demand, as well as other numerous technologies adaptations. Hence, many transmitted signals or packets got dropped or corrupted before reaching their desire receivers, while some got delivered has errors with imperfect values, hence lessening the quality of service (QoS). [1]. Another issue is the demand for electromagnetic spectrum (i.e. Radio Frequency (RF), which is a valuable resource, limited as well as underutilized. However, on a continuous basis, RF demand has been exponentially rising [2]. Hence, a good schedule and an effective management are required towards enhancing the WCS's congested environment.

The CR is made up of a combination of an embedded hybrid intelligence module called Cognitive Engine (CE) and a Software Defined Radio (SDR). The CE is the essential component of CR through which its self-organizing capabilities are been carried out and comprises of a knowledge base, a reasoning, and a learning engine [3]. The reasoning engine is considered to be the CE's central processing unit which controls every status and performs the logical thinking. Meanwhile, the learning engine learns every CR dynamical adjustments or settings based on its previous system responses. The knowledge-based engine maintains and keeps every tracks or event information within the CR system [4]. Originally envisioned by Mitola [5] and later adopted by Haykin [2], CR is operated on a looped-intelligence sequence called cognition cycle as illustrated in Fig. 1. Through this procedure, CR is able to sense the WCS environment, plan its communication actions according to input received from sensors and create the required network policies, and at the same time learn from its previous results, before it finally decides.

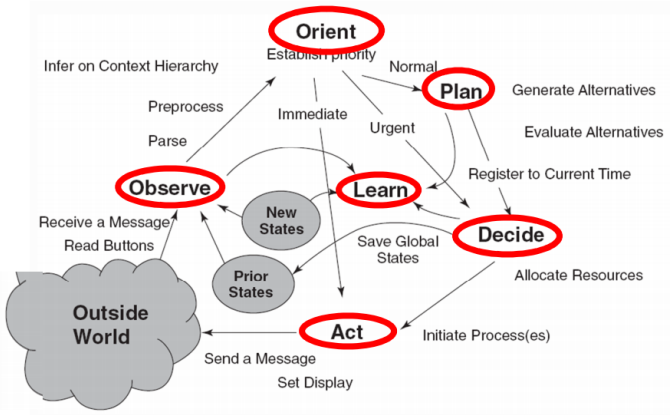

Fig. 1: - The Cognitive Cycle [2]

Cognitive Radio (CR) provides solution to various WCS's limitations, most especially controlling the wireless communication unstable communication medium (i.e. spectrum). CR can be applied to any WCS's communication services, and networks that suffer from spectrum shortage and any congested issues. Because CR is developed has an intelligent device and effectively considered to be the required promising solution to WCS improvement. CR uses this intelligent capability to handle the spectrum management in an opportunistic manner. And, once the CR selects any available channel, the next challenge is to make the network protocols adaptive to the available spectrum, using its awareness with its wireless environment, ability to learn from its past operations and being

able to automatically adjust its radio operating parameters (e.g. modulation, transmit-power, carrier-frequency, and other communication policies) to adapt to the incoming RF stimuli in real- time. [1].

CR intelligence is called Cognitive Engine (CE), and comprises of three separate components, namely: reasoning engine, learning engine and decision engine. Out of all the three CE's intelligence components; the learning engine has been selected has the primary focus of this paper, by proposing an Artificial Intelligent system based on Random Neural Network (RNN) has an alternative learning engine for a CR intelligence engine. In the summary, by adapting RNN's learning attributes, it is believe that the CR will be able to perform the followings: reduces CR development complexity, speed up the communication processes, time and equally generate better and a more reliable decision.

Next, a discussion on RNN is prompted to introduce readers in the application of RNN's learning competency and cognitive radio engine development. The need for an improved learning system become necessary, especially in regards to the existing ANN learning system. For instance, ANN could not properly fit in heavy and noisy data, which can be considered to be equivalent to the present noisy WCS. And for this reason, RNN learning capacity is hereby proposed has a better alternative. Although, learning in neural network systems, has been found out to be an elementary mathematical operation. Whose output structure is heavily dependent on the activation function [3].

RNN models are tractable, reliable in performance, and can easily be implemented in both software and hardware. However, its neurons spiking potential features will be critically observed and adapted to implement our proposed CEs learning engine; because its output is a little closer to that of human representation, in terms of its prediction and accuracy better than that of ANN [6]. For example, RNN system has been used in classification problems, intelligent recognition solvers, such as coloured images recognition plus in solving various optimization problems, with lots of RNN literature and applications presented and discussed in [7], However, the entire NN's capabilities are yet to be fully exploited in the CR development, for the fact that there are still some pending issues to clarify, such as faster decision making, accurate users performances, and precise predictive results.

Principally on the real-time wireless environment, this has become more pronounced and is getting more complicated, especially among the heterogeneous CRNs; where a CR not only has to adapt to the RF environment, but is expected to coordinate its operations with other radios as well as within different networks. In solving the aforementioned WCS related problems, precisely congestion control and the unpredicted wireless environmental predicaments, such as faster decision making, accurate users performances, and precise predictive results. To achieve this, the main scope of this paper has been shifted towards developing an optimized computer generated simulation based learning algorithm using the RNNs learning properties.

The rest of the paper is structured as follows: related work and motivation for application of random neural networks to cognitive radio systems are presented in Section II. Section III introduces a revisiting Random Neural Network (RNN) mathematical expression. The modelling approach through which the proposed objective was realised was presented in Section IV. While, the RNN training procedure is expressed in Section V, and Section VI discusses the simulation results, with the paper been concluded in Section VII.

II. Related Works

He *et al.* in [8], used a case-based reasoning method to design and develop a CE learning engine for IEEE 802.22 wireless regional area network (WRAN). In [9], ant colony optimization learning algorithm was implemented to manage multi-channel radios communication. A completely different CR learning model technique was adopted by Newman [10] and Rieser [11] for the dynamic resource allocation in the congested WCS. Other AI solutions include the use of ANN based CR learning system and large scale WCS optimization by Hasegawa in *et al* [12] and Baldo in [13]. Another CE implementation is presented in [14] and involve two intelligence combination of Genetic Algorithm (GA) and Radial Basis Function Neural Network. The scope of this paper was adopted with an associated RNN based learning representation known has "A linear non-negative least square (NNLS) learning system" and was applied in [15] in solving a related unstable problem in a disaster condition. Although, an extensive learning improvement was carried out in [16] and [17]; and various modifications made to the RNN neural structure through a large ensemble of cells, until synchronized interactions exist between its neurons to trigger asynchronous firing and its outcome really shows an impressive improvement to RNN's application. In addition, several CR learning engines had been implemented with other AI techniques, such as: metaheuristic algorithms, hidden Markov models, rule-based reasoning, ontology-based reasoning, and case-based reasoning. For the readers conveniences [18] and [19] are recommended.

III. Random Neural Network Mathematical Model

Like ANN neural modelling, RNN also exist in two types, namely: the feed-forward and the recurrent neural systems. The feed-forward architecture was used in for our proposed RNN-CE learning algorithm, and is diagrammed in Fig. 2. It is a three layer neural model comprising of an input layer with our seven defined variables, the hidden layer of which six (6) nodes was used in the simulation and an output layer with only one neuron determining the available throughput.

RNN model is made of interacting N-neurons with two internally generated signals, which are excitatory and inhibitory signals. However at a state, these interacting N-neurons get

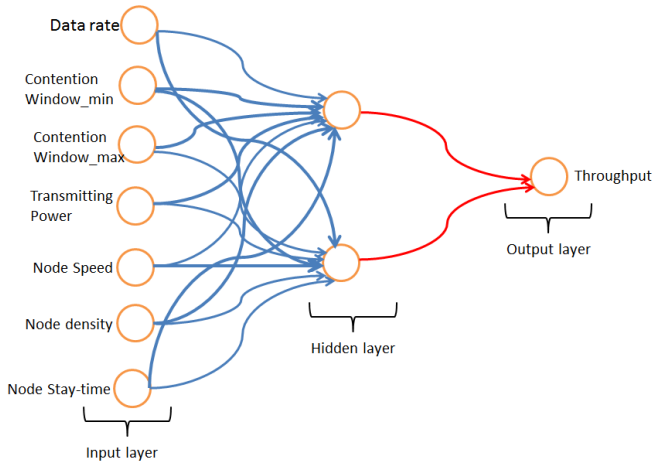

Fig. 2: - Random Neural Network Modelling

engaged and circulate among each other at a specific time(t) to generate a form of spikes energy called potential $K(t)$ and mathematically expressed in (1).

$$K(t) = (K_1(t) + K_2(t) + K_3(t)..... + K_n(t)) \quad (1)$$

where: $K_i(t)$ is the potential value of neuron(i).

If the potential $K_i(t)$ of neuron(i) produced is positive, then, the neuron(i) is said to be excited (excitatory state). Meanwhile, if it is a negative signal, it is considered has inhibitory [20]. And the rate at which each neuron fires, is called constant Poisson process rate (r_i). However, every time this occurs, the RNN network steady-state probability changes from q_i state to $q_{(i-1)}$, and as a result reduces its potential $K(t)$. If the arriving signals are negative, the potential $K(t)$ produces will be reduced by 1, while on the contrary, the potential $K(t)$ will get increases by 1 for every positive signal arrival [21] and [22]. The probability that a neuron(i) generates and sends positive signals to neuron(j) is denoted as $p^+_{(i,j)}$, while the equivalent probability for negative signals is expressed as $p^-_{(i,j)}$. However, probability of a neuron(i) signal leaving or departing from the network is represented as d_i. While based on the Markovian theory [23], the N number of neurons movement probabilities was developed and mathematically expressed in (2):

$$d_i + \sum_{j=1}^{N} \left(p^+_{ij} + p^-_{ij}\right) = 1 \qquad \text{for } 1 \le i \le N \quad (2)$$

The transition probability $p_{(i,j)}$ representing the movements of signals between neurons is represented by (3)

$$p_{(i,j)} = \left(p^+_{ij} + p^-_{ij}\right) \quad (3)$$

The combination of the probabilities in (2) and (3), presents an approach through which RNN synaptic weights (w) can be determined for both positive by (4) and negative signals by (5).

$$w^+_{(i,j)} = r_i p^+_{(i,j)} \ge 0 \quad (4)$$

$$w^-_{(i,j)} = r_i p^-_{(i,j)} \ge 0 \quad (5)$$

where (w) is the weight and equivalent to the ANNs, only that the RNN synaptic weights are non-negative and used for the excitatory and inhibitory spike rates. The r_i in (6) is the firing rate between the neurons and regarded as a Poisson firing rate with exponential distributed interim pulse intervals.

$$r_i = \sum_{j=1}^{N} \left[w^+_{(i,j)} + w^-_{(i,j)}\right] / \left(1 - d(1)\right) \quad (6)$$

While, the steady-state probability q_i for an excited neuron [7], [21], and [20] is expressed with (7) .

$$q_{(i)} = \left[\frac{\lambda^+_{(i)}}{r_{(i)} + \lambda^-_{(i)}}\right] \quad (7)$$

where, $\Lambda_{(i)}$ denotes the Poisson process of rate at which external excitatory (positive) signals arrived into the neuron(i) and $\lambda_{(i)}$ is the equivalent external inhibitory (negative) signal into neuron i. The total external arrival rate for the positive λ^+_1 and negative λ^-_1, are regarded as non-linear simultaneous equations, and respectively expressed in (8) and (9).

$$\lambda^+_{(i)} = \sum_{j=1}^{n} \left[q_{(j)} r_{(j)} p^+_{(j,1)}\right] + \Lambda_{(i)} \qquad for 1 \le i \le N \quad (8)$$

$$\lambda^-_{(i)} = \sum_{j=1}^{n} \left[q_{(j)} r_{(j)} p^-_{(j,1)}\right] + \lambda_{(i)} \qquad for 1 \le i \le N \quad (9)$$

At time t, the vector of the generated potential $K(t)$, such that $(K(t) = (K_1(t) + K_2(t) + K_3(t)..... + K_N(t)))$ determine the state of the network. The stationary probability distribution $p(k)$ then expressed by (10),

$$p\,(k) = \lim_{t \to \infty} prob[k\,(t) = k] \quad (10)$$

However, if (8) and (9) produces a non-negative solution, such that $q_i \le 1$, then the stationary probability distribution $p(k)$ solution becomes (11)

$$p_{(k)} = \prod_{1=1}^{n} \left[1 + q_i\right] q_i^{k_i} \quad (11)$$

At this point, the network become stable and guarantees the excitation status for each neuron of remaining finite with probability = 1. Similarly, the average potential A_i of a neuron(i) can be determine as

$$A_i = \frac{q_i}{(1 - q_i)}$$

If the steady-sate probability q_i in (7) is greater than zero, i.e.,

$$q_{(i)} = \left[\frac{\lambda^{+}_{(i)}}{r_{(i)} + \lambda^{-}_{(i)}} \right] > 0$$

then the neuron(i) is considered to be saturated or unstable and continuously firing in steady state.

The RNN symbols used in this paper are presented in Table I.

TABLE I: The list of RNN notations with their meaning

$k_i\,(t)$	Potential of neuron i at time t
q_i	Probability that neuron i is excited in steady state
Λ_i	External arrival rate of +ve signals to neuron i
λ_i	External arrival rate of -ve signals to neuron i
$\lambda^{+}_{(i)}$	Average arrival rate of +ve signals to neuron i
$\lambda^{-}_{(i)}$	Average arrival rate of -ve signals to neuron i
$w^{+}_{(1,j)}$	Rate of +ve signals to neuron j when neuron i fires
$w^{-}_{(1,j)}$	Rate of -ve signals to neuron j when neuron i fires
d_i	Probability that a signal from firing neuron i departs from the network
r_i	Firing rate of neuron i

IV. Proposed Approach

Has illustrated in Fig-3, describing the simulation approach through which the scope of this work was designed, modelled and implemented. The starting point was the installation for the Network Simulation version-2 (NS-2) [24] on a linux based operating system and patching of all the relevant as well as related files. After this, a Cognitive Radio Network (CRN) was developed with a multi-radio multi-channel physical layer using an Ad-hoc On-Demand Distance Vector (AODV) routing protocol. AODV, is a special routing protocol programming developed for mobile ad hoc networks (MANETs) and other mobile wireless ad-hoc networks.

Next was the Software Defined Radio (SDR) parameter configuration using the TCL script, and in turn used to develop the comprehensive dataset. To achieve this, the two classes of CR's parameter settings was modelled, which are the transmission and environmental parameter settings has listed in Table II and III. These two sets of parameters are predefined, varied and manually in-putted into the TCL script to execute different wireless scenarios. After generated the dataset, the collated values matched with corresponding simulation output results, which are the modelled CRN available throughput. And for the final step has displayed in the Fig-3, there are two NN models proposed. The Matlab's Neural Network (NN) toolbox package [25] was used to implement the Artificial Neural Network (ANN) learning engine, while the proposed RNN learning engine implementation was performed with another

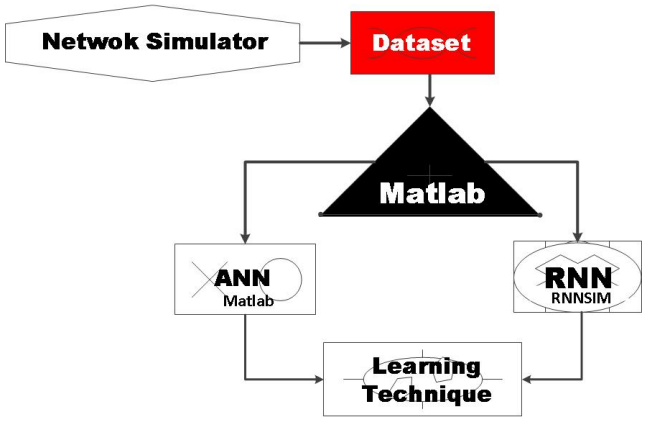

Fig. 3: - The Simulation Implementation Layout

third party NN package named Random Neural Network Simulator (RNNSIM) toolbox [26].

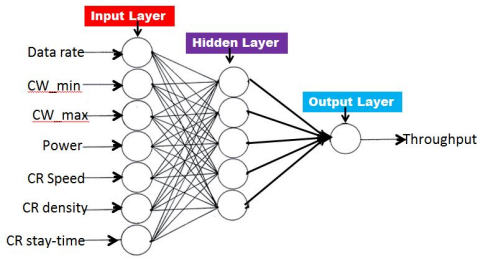

Fig. 4: - Artificial Neural Network Modelling

Each of the generated results from various different WCS environmental condition is manually defined to generate the entire 480 datasets, has articulated in Table IV. Combining the CR two parameter classes, give seven (7) operational parameters in total were used to generate the datasets, has articulated in Table IV. All these seven parameters were collated and equally used has inputs for the NN learning modelling structures has respectively diagrammed in Fig 2 and Fig 4 for both RNN and ANN systems. With the transmission set having four variables and three (3) unsteady environmental conditions. Various tunings were made to the two NN learning modelling structures, such as the number of hidden layer neurons, the minimum square error , and the correlation factor between the expected and the predicted CRN available throughput in order to validate the two learning techniques performances.

TABLE II: Proposed CE Transmission Parameters

Protocol	**Data rate**	**Power**	**CW-min**	**CW-max**
AODV (R)	1Mbps	1P	31	1023
DSDV	2Mbps	2P	63	2047
WCETT	5.5Mbps	3P	.	
	11Mbps	.		
				(P=0.28183815W)

TABLE III: CRN Network Parameters

Scenario:1	**Scenario:2**	**Scenario:3**
CR speed (V)	Number of CR node (N)	CR Stay-time (Pt)
10km/h	5	0s
15km/h	8	5s
20km/h	12	10s
30km/h	15	20s
40km/h	20	35s
50km/h	25	50s
	35	100s
CR nodes=20	CR Speed=20Km/h	CR nodes=20
Stay-Time=10s	Stay-Time=15s	Speed=20Km/h

The transmission cycle (TC) was defined in (12) as the product of the proposed transmission variables in Table II.

$$TC = N_{DR} \times N_{CW} \times N_P \tag{12}$$

where:
N_{DR} = the number of data rate = **4**
N_{CW} = the number of contention windows = **2**
N_P = the number of transmitting power = **3**.

This in total gives **24** different wireless environmental settings. However, to generate additional scenarios, TC was later combined with Table III to produce Table IV; which is the over CRN simulation conditions executed.

TABLE IV: CE Network Setting Calculation

CR Node speed	TC * $CR{-}_{speed}$	24 * 6 = 144
CR Node number	TC * $CR{-}_{number}$	24 * 7 = 168
CR Node Staytime	TC * $CR{-}_{stay-time}$	24 * 7 = 168
Total = 144+168+168 = 480 CRN Simulation Scenarios		

where:
$CR{-}_{speed}$ = Number of speed scenarios,
$CR{-}_{number}$ = Number of CR density scenarios
$CR{-}_{staytime}$ = Number of scenarios for stay-time

V. Training Implementation Design

The purpose of supervised learning is to find an optimal set of weights so that for a set X of input patterns, the AI models output closely represents the desired one. The RNN non-negative least X-th pattern was formulated has NK form equation in (13), by combining all the RNN excited steady state in (7) with all its known preferences and assumptions for (8) and (9), and its constant firing rate r_i in 6 in section III above.

$$q_{ik} \sum_{N}^{j=1} w^{-(i,j)} + q_{ik} \sum_{N}^{j=1} q_{jk} w^{-(i,j)} + q_{ik} \sum_{N}^{j=1} w^{+(i,j)} - q_{ik} \sum_{N}^{j=1} q_{jk} w^{+(i,j)} = \Lambda_{iki} - q_{ik}\lambda_{ik}, \forall_i, k \tag{13}$$

Such that: $w^{+(i,j)} \geqslant 0, \quad \& \quad w^{-(i,j)} \geqslant 0 \quad \forall, \quad i, \quad j$

For equation (13) becoming linear with nonnegativity constraints, then all q_{ik} must be determined (i.e. must be known). Meanwhile the 2 weights in 8 and 9 becomes unknown due to various adjustments within the NN model to produce a closely related predicted output from its supplied original data. The assumption that all q_{ik} are known is valid because q_{ik} can be set to random values. To achieve a better and accurate learning result, a 10-fold K-cross validation was used to split the comprehensive dataset into two separate datasets, which are the training and testing datasets. The entire dataset of 480 results was partitioned into 10 segments to produce different 48 segments of data . And after averaging each of these segments, a portion was selected and was used as the testing data, while the remaining nine segments were pulled together to produce the training dataset. Although the training time was shorter, the accuracy achieved was great, because the true error rate estimator was small. However, the computational time is longer.

Traditionally, hidden layers plays a vital role in the NN performance, especially in the case where problems related to arbitrary decision to arbitrary accuracy. Typically, hidden layer is considered as a black box, as it is essentially hidden from view. The hidden layer introduces greater processing power and system flexibility into the NN model, although it comes with additional complexity on the training algorithm. The process or rules for determining the number of neurons for the hidden layer has still not yet been certain. Although, having too many neurons in the hidden layer will have the system over-specified, and unable to be generalized. However, hidden layer with moderate or few neurons make the system simple and easy to extrapolate its data, and equally preventing the network from being poorly trained or over-fitting [3], [8] and [12].

Parten et al. in [27], presented an assumption, expressing the maximum hidden layer neuron numbers (HN_{Max})) as:

$$(N_{Input}) + (N_{Output}) \text{ x } 1$$

While, the minimum neuron in the hidden layer (HN_{Min})

$$(2 \text{ x } N{-}_{Input}) + N{-}_{Output})$$

where:
$N{-}_{Input}$ = the number of input variables and
$N{-}_{Output}$ = is the number of output variables

VI. Results Analysis

In this section, simulation results are presented and analysed, by comparing the proposed RNN-CE learning engine with the ANN-CE learning engine. Once the two engines got structured and modelled, they were trained equally using the same dataset, training algorithm (gradient descent) and learning rate. Training begins when the connecting neuron weights are randomly selected and initialised. Training in NN is classified into two categories; supervised and unsupervised [13], [14], [21], with us using the supervised training.

A. Performance Criteria

To evaluate the two learning engines' performance, two separate approaches were considered the numerical (presented in Table V) and the graphical representations in Figures 6 and 7. These two approaches were achieved through two standard performance estimator functions, namely the Mean Square Error (MSE) and the Regression Factor (R). The MSE [14] is expressed in (14), and the Regression Factor (R) is defined in (17):

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(O_i - T_i)^2 \tag{14}$$

where:
O_i = the desired (Network predicted) outputs
T_i = the network outputs (Actual supplied output)
N = the total number of dataset.

The Regression Factor (R) is a statistical analysis establishing the relationship between two variables. In this paper the regression factor will be used to evaluate the correlation between the proposed learning engine output and the actual supplied output. The mathematical expression of R is given as:

$$y = Mx + B$$

where:
y = the actual output (Target)
x = the predicted output
M = the regression coefficient
B = the constant (sometimes referred to as random error)

The M is determined using the least squares method, by minimizing the sum of the squares of the differences between the actual output values and the predicted learning engines output values. Similarly, the M is the slope of the regression graph, while the B is the intercept.

$$M = \frac{\sum x_1 y_1 - \frac{\sum x_1 \Sigma x_1}{n}}{\sum x_1^2 - \frac{(\sum x_1)^2}{n}} \tag{15}$$

$$B = \frac{\sum y_1 - b\Sigma x_1}{n} \tag{16}$$

The Regression Factor (R) between the x and y variables is given by:

$$R = \frac{\sum x_i y_i - \frac{\Sigma x_i y_i}{n}}{\sqrt{\left(\Sigma x_i^2 - \frac{(\sum x_i)^2}{n}\right)\left(\sum y_i^2 \frac{(\sum y_i)^2}{n}\right)}} \tag{17}$$

To validate the network performance, the obtained training results are compared with the two learning algorithms. The result with the lowest MSE and highest Regression Factor (R) is considered to be the best. A regression factor value closer to 1 (i.e. R=1), indicates the predicted network output is establishing a strong correlation with the actual output. As described in [13], [25], the training was terminated when the validation set attained the set minimum MSE value, from the numerical approach, as observed in Table V. The proposed RNN-CE learning algorithm was able to establish its strong prediction and generalization abilities. It converges at 350 epochs with MSE value of 0.000268, while ANN learning algorithm took 2640 iterations to achieve MSE value of 0.00081588.

TABLE V: Numerical Analysis

Performance Statistic	**RNN**	**ANN**
MSE for training set	0.000268	0.00081588
Regression Value	0.95188	0.86877
Hidden node neurons	6	10
Number of epochs at the end	350	2640
Learning Algorithm	Gradient Descent	Gradient Descent

Applying the simulation parameters into Parten et al. [27] assumption, the expected minimum and maximum number of neurons in the hidden layer becomes:

HN_{-Min} = (7+1) x 1 = 8 neurons
HN_{-Max} = (2 x 7) + 1 = 15 neurons

By Parten's calculation [27], the expected minimum number of neurons in the hidden layer was 8 neurons, and 15 neurons as the maximum. However, RNN model was able to extrapolate with 6 neurons, while ANN model hidden layer was 10 neurons .

B. Generated User's Communication Scenarios

Three (3) datasets in Table VI were picked from the testing segment to evaluate the training performance against the two NN models and the NS-2 generated results. The outcome is displayed in Fig 5.

The graphical representations for the two learning algorithms' performances are presented in Figures 6 and 7. Figure 6 displays the training data set performance for three outputs: the proposed RNN-CE learning engine, the ANN-CE learning engine and the actual output (the generated CRN throughput). The RNN-CE learning engine predicted output was closely

TABLE VI: Generated Scenarios

PARAMETERS	User - 1	User - 2	User - 3
Data-Rate (MB)	11	5.5	2
Power (W)	0.3805	0.3523	0.31002
User-density	10	15	30
Speed (Km/h)	20	30	10
RESULTS	**Available Throughput**		
NS-2	478.45	365.59	251.6
RNN	445.31	332.12	224.75
ANN	405.08	312.68	201.85

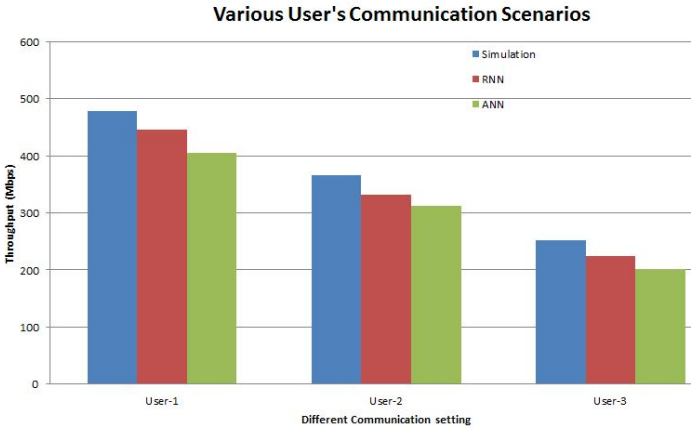

Fig. 5: Learning Testing Performance Comparison

mapped to the actual CRN output compared to the ANN predicted output. This implies the ANN-CE learning engine predicted CRN throughput value was unable to attain the actual output. But the RNN-CE learning engine achieved about 93% closer output result to the original supplied CRN throughput output. In Figure 7, the correlation relationship between the two learning engines and the actual output were presented. The RNN learning algorithm has a Regression Factor (R) of 0.95188, with 8% error margin away from the actual output. Meanwhile, (R) recorded for the ANN engine predicted output was 0.86877, making it 8% errors away from the original output, plus 30% reduction in its number of the hidden neurons used (from ANN using 15 neurons to RNN with only 8 neurons). Hence, the RNN is able to predict accurately at 4.6% higher than the ANN learning engine.

VII. Conclusions

In this study, we presented an AI based learning framework towards improving the wireless communication systems as well as addressing the rate of decision making in Cognitive Radio Network. A Random Neural Network based on the Nonnegative Least Squares Supervised (NNLSS) learning was implemented and extensively verified using two separate computer-based learning algorithms. The two models are subjected to the same conditions, i.e. same WCS environmental and same learning premises. The overall performance expressed in table VII, shows that the RNN based learning algorithm was 36.895% better than the standard ANN learning engine. With great predicted output and faster learning execution (i.e.e processing time). The future research directions include a critical and a broader evaluation of the training process, by generating a larger dataset through additional wireless communication operational parameters, as well as using another supervised learning method with one or two additional protocols.

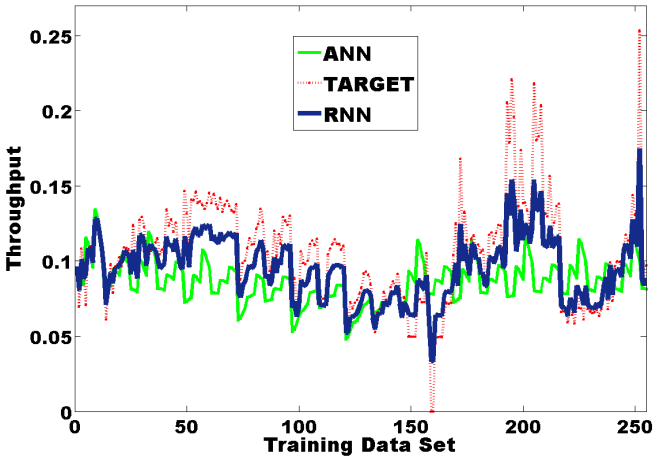

Fig. 6: Learning Prediction Performance Comparison

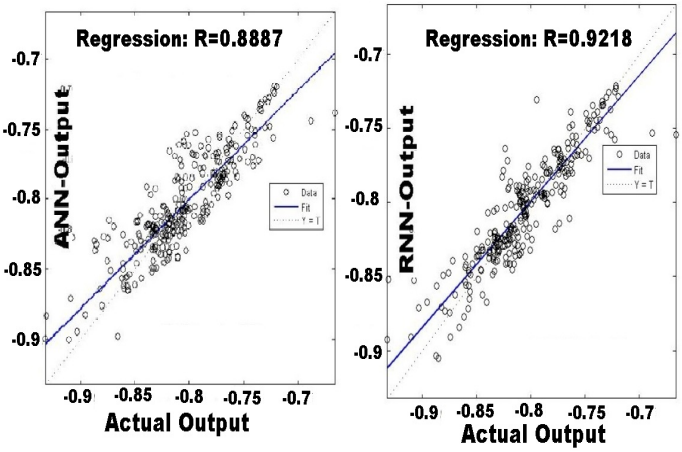

Fig. 7: Learning Performance Correlation Output

TABLE VII: Cumulative Simulation Performance Results

Function	RNN	ANN
Training MSE	24.73%	75.27%
Regression	52.28%	47.72%
Hidden Nodes	37.5%	62.5%
Epochs	11.7%	88.3%
Cumulative Result	**31.55%**	**68.45%**

Acknowledgement

The authors would like to thank the management and staff of Petroleum Technology Development Fund (**PTDF**), Nigeria for their support toward this study.

REFERENCES

[1] Akyildiz, Ian F. Lee, Won-Yeol, Vuran, Mehmet C. and Mohanty, Shantidev, "NeXt generation dynamic spectrum access cognitive Radio Wireless Networks:A Survey," Elsevier Journal, vol. 50, pp. 2127-2159, May 2006.

[2] S. Haykin, "Cognitive radio: Brain-empowered wireless communications," IEEE wireless Communications Journal, vol. 23, pp. 201220, Feb. 2005.

[3] C. Clancy and J. Hecker and E. Stuntebeck and T. O'Shea, "Application of machine learning to cognitive radio networks," IEEE Wireless Communications, Vol. 14(4): pp. 47-52, Aug. 2007.

[4] Alexander M. Wyglinski, Maziar Nekovee, Y. Thomas Hou, "Cognitive Radio Communications and Networks (Principles and Practice)," Academic Press, Cambridge, Massachusetts, USA. Nov. 2009.

[5] J. Mitola and G. Q. Maguire, "Cognitive radio: making software radios more personal," IEEE, ON: vol.6, no.4, pp.13-18, Aug 1999.

[6] Aguilar Jose, "Definition of an energy function for the random neural to solve optimization problems," Neural Networks, vol. 11, pp.731738, Jun. 1998.

[7] Stelios Timotheou, "The Random Neural Network: A survey," Computer Journal Advance Access, vol. 53, no 3 Apr. 2010.

[8] An He, Kyung Kyoon Bae, Timothy R. Newman, Joseph Gaeddert, Kyouwoong Kim, Rekha Menon, Lizdabel Morales-Tirado, James Jody Neel, Youping Zhao, Jeffrey H. Reed, and William H. Tranter, "A Survey of Artificial Intelligence for Cognitive Radios," IEEE Vehicular Technology, vol.59, no.4, pp.1578-1592, May 2010.

[9] Nan Zhao, Shuying Li and Zhilu Wu, "Cognitive Radio Engine Design Based on Ant Colony Optimization," Wireless Personal Communications on, vol. 65, no. 1, pp 15-24, Jul. 2012.

[10] T. R. Newman and J. B. Evans, "Parameter Sensitivity in Cognitive Radio Adaptation Engines," New Frontiers in Dynamic Spectrum Access Networks, 3rd IEEE Symposium, pp. 1-5, Oct. 2008.

[11] Rieser, Christian James, "Biologically Inspired Cognitive Radio Engine Model Utilizing Distributed Genetic Algorithms for Secure and Robust Wireless Communications and Networking," PhD. Dissertation. 2004.

[12] Hasegawa, Mikio and Takeda, Taichi and Kuroda, Taro and Tran, Ha Nguyen and Miyamoto, Goh and Murata, Yoshitoshi and Harada, Hiroshi and Kato, Shuzo, "Application of Higher Order Neural Network Dynamics to Distributed Radio Resource Usage Optimization of Cognitive Wireless Networks," Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.

[13] N. Baldo and M. Zorzi, "Learning and Adaptation in Cognitive Radios Using Neural Networks," IEEE Consumer Communications and Networking Conference, Jan. 2008.

[14] Y. Yang and H. Jiang and C. Liu and Z. Lan, "Research on Cognitive Radio Engine Based on Genetic Algorithm and Radial Basis Function Neural Network," Engineering and Technology (S-CET) Spring Congress, May 2012.

[15] Stelios Timotheou, "Nonnegative Least Squares Learning for the Random Neural Network," Artificial Neural Networks, pp. 195-204, Jan. 2008.

[16] Erol Gelenbe and Stelios Timotheou "Synchronized interactions in spiked neuronal models," The Computer Journal, vol. 51, no. 6, pp. 723-730, Mar. 2008

[17] Erol Gelenbe and Stelios Timotheou "Random neural networks with synchronized interactions," Neural Computation, 20, pp. 23082324, 2008.

[18] A. He and K. K. Bae and T. R. Newman and J. Gaeddert and K. Kim and R. Menon and L. Morales-Tirado and J. '. Neel and Y. Zhao and J. H. Reed and W. H. Tranter, "A Survey of Artificial Intelligence for Cognitive Radios," IEEE Transactions on Vehicular Technology-, vol.59, no.4, pp.1578 -1592, May 2010.

[19] Michael Georgiopoulos and Cong Li and Taskin Kocak, "Learning in the feed-forward random neural network: A critical review," Performance Evaluation Journal, vol. 68, no. 4, pp.361384, 2011.

[20] Erol Gelenbe, "Learning in the Recurrent Random Neural Network," Neural Computation Journal, Vol. 5, no. 1, pp.154-164, Jan. 1993.

[21] Aguilar Jose, and Molina Cristhian, "The Multilayer Random Neural Network," Neural Processing Letters, vol. 37, pp. 111–133, 2012.

[22] Aristidis Likas. and Andreas Stafylopatis, "Training the Random Neural Network Using Quasi-Newton Methods," European Journal of Operational Research, vol. 126, no. 2, pp 331-339, 2000.

[23] Rangarajan.A, Chellappa, and Manjunath.B.S, "Markov Random Fields and Neural Networks Applications," Artificial Neural Networks and statistical Pattern Journal, 1991.

[24] Issariyakul, Teerawat and Hossain, Ekram, "Introduction to Network Simulator NS2," Springer Publishing Company, USA, 2011.

[25] MATLAB, "Version 7.10.0 (R2014b)," The MathWorks Inc.,Natick, Massachusetts, 2010.

[26] H. Abdelbaki, "Random neural network simulator in Matlab," ftp://ftp.mathworks.com/pub/contrib/v5/nnet/rnnsimv2/.

[27] Parten, C. , et al., "Handbook of Neural Computing Applications," Academic Press, San Diego, California, 1990.